csci 3411: Operating Systems

Protection

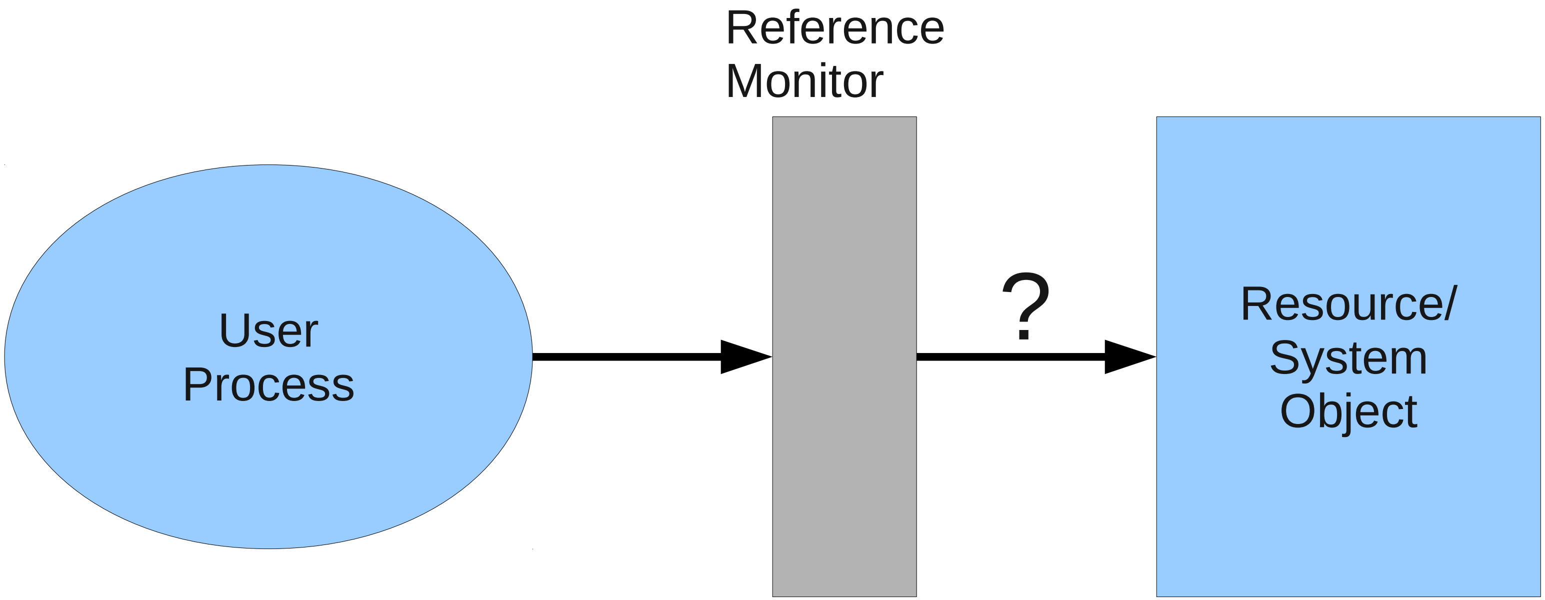
Gabriel Parmer

Slides evolved from Silberschatz and Stanton

Protection

- System consists of collection of resources
  - Physical resources
    - Memory, Disk, and NIC
  - Virtual resources
    - Files, Processes, Semaphores
- System has a number of *principals*
  - Entities that access the system resources
  - Users, Processes, Threads
- Protection: ensure all resources accessed correctly, and only by those principals that are allowed to do so

Access Control



- System knows who the requester of the resource is (the principal)
- All accesses to resources go through the reference monitor
  - Can the requester access the resource or not?

Access Control II

- Reference monitor
  - Is trusted
  - Must be protected itself from principals
- How does the reference monitor decide if a principal should access a resource?
  - Guiding principles
    - Principle of least privilege (POLP)
    - Need to know
    - IITYIHTKY

Access Matrix

	File 1	File 2	File 3	Network	Printer
User 1	read	write	-	-	print
User 2	write	execute	write	receive, send	-
User 3	-	-	-	receive, send	-
User 4	read, execute	write	-	send	-

- Principal *i* allowed to perform operation *op* on resource *j* if *op* є *AM(i,j)*

Access Matrix II

- Mechanism
  - How does the reference monitor ensure that all executed operations are allowed by the access matrix?
- Policy
  - How are the specific access rights for objects placed into the access matrix?
- General Goal: Make a general mechanism that can support the largest variety of useful policies

Protection Mechanisms: Table

- Global Table – Access matrix stored as large table in memory and on disk
  - Size = # principals * # resources
    - # users in engineering?
    - # files on a file system?
  - Principals can include processes
    - Must add principal when process is forked
    - Remove principal when process exits

Protection Mechanisms: ACLs

	File 3
User 1	-
User 2	write
User 3	-
User 4	-

- Corresponds to access matrix columns
- Access Control Lists (ACLs)
  - Each resource has a list associated with it (metadata)
  - For files, ACLs stored in filesystem
  - Every time a principal attempts an operation on a resource, check if ACL gives access

Protection Mechanisms: Capabilities

	File 1	File 2	File 3	Network	Printer
User 2	write	execute	write	receive, send	-

- Capabilities correspond to access matrix rows
- Access rights for resources associated with specific principals
  - User 2 has a capability to write to File 1
  - Ownership of a capability for an operation to a resource is designation of right to access
    - Reference monitor simply checks for presence of capability
    - Capabilities cannot be directly accessible/modifiable – trusted

ACLs and Capabilities

- Bolt Bus vs. DC2NY
  - Ticket as proof of entry
    - Don't even need to know passenger name
    - Different levels of access
  - List of passengers
    - Accessed for each arriving passenger
- DC2NY has two employees/bus; Bolt Bus, one
  - Which uses ACLs, which capabilities?

ACLs and Caps: Comparison

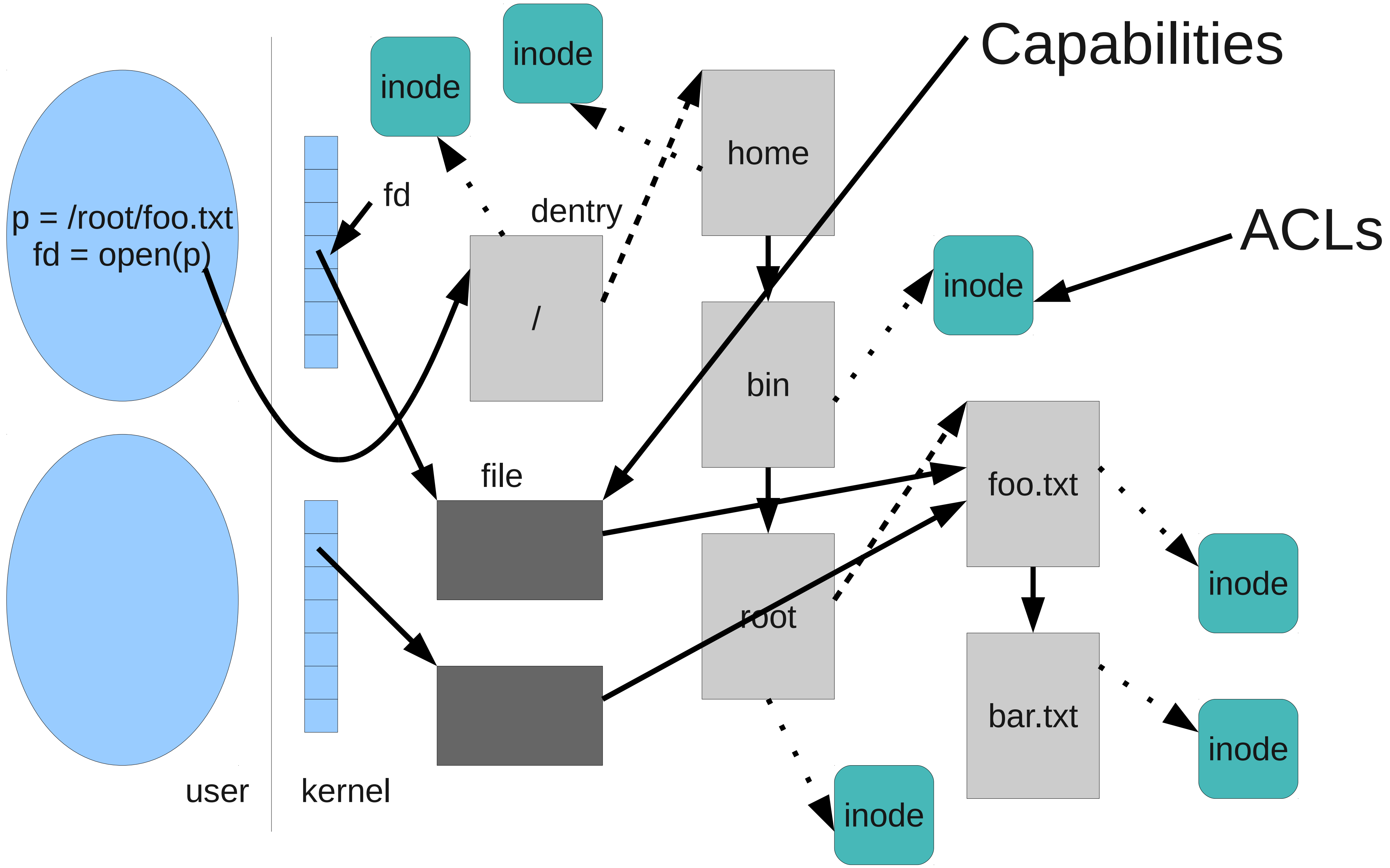
- Reference Monitor
  - ACLs: checking lists can be time-consuming
  - Caps: presence of cap designates access – fast!
- Delegation – give access rights to other principal
  - ACLs: access/modify ACL
  - Caps: pass capabilities to other principals at runtime
- Revocation – remove previously granted rights
  - ACLs: remove principal's access from ACL
  - Caps: Difficult (track all capabilities, level of indirection,)

Often Complementary Mechanisms

- Drink bracelet at concerts
  - To get bracelet, expensive check of “list”/wallet
  - Once have bracelet, cheap verification of age

- open vs. read/write
  - open traverses filesystem, checking access
  - File descriptor denotes ability to access the file
    - Capability that precludes expensive access control checks

Caps and ACLs



Policies: Bell-LaPadula

- Information flow: if we treat data as if it went to all places in the system permitted, does anyone see it who shouldn't?
  - Assume: my password is written on my desk
  - Qi has a key to my office
  - Elizabeth has access to Chonti's desk
- Bell-LaPadula specifies
  - Classification of data and users into levels
  - How information can flow between the users based on the levels of the data

Bell-LaPadula Confidentiality

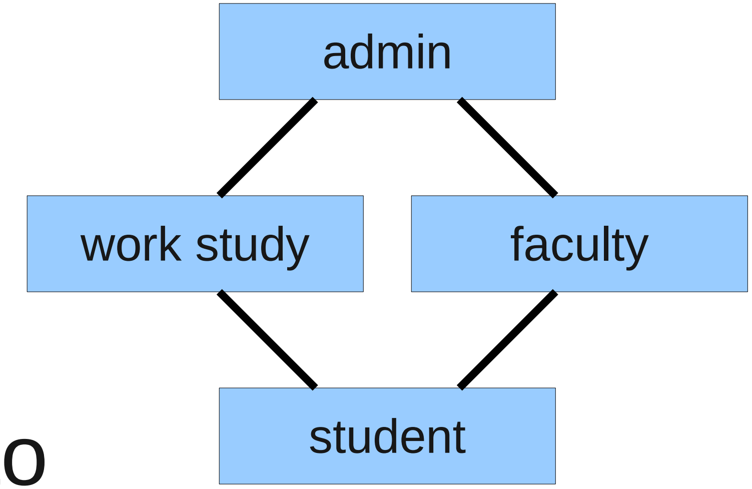
- Assume: *C(x)* is the classification level of resource or user *x*
- Simple Security Property:
  - For user *u*, resource *x*, *u* may read *x* if

    $C(x) \leq C(u)$
- *-Property:
  - A principal with read access to *x* may write *y* if

    $C(x) \leq C(y)$

Policies: Role Based Access Control

- Role – set of users
  - Assign access permissions to roles, not users
  - Users get all permissions from the rule their in
  - Partial order of roles
    - Role gets permissions of all roles below
    - Only list new permissions at a level

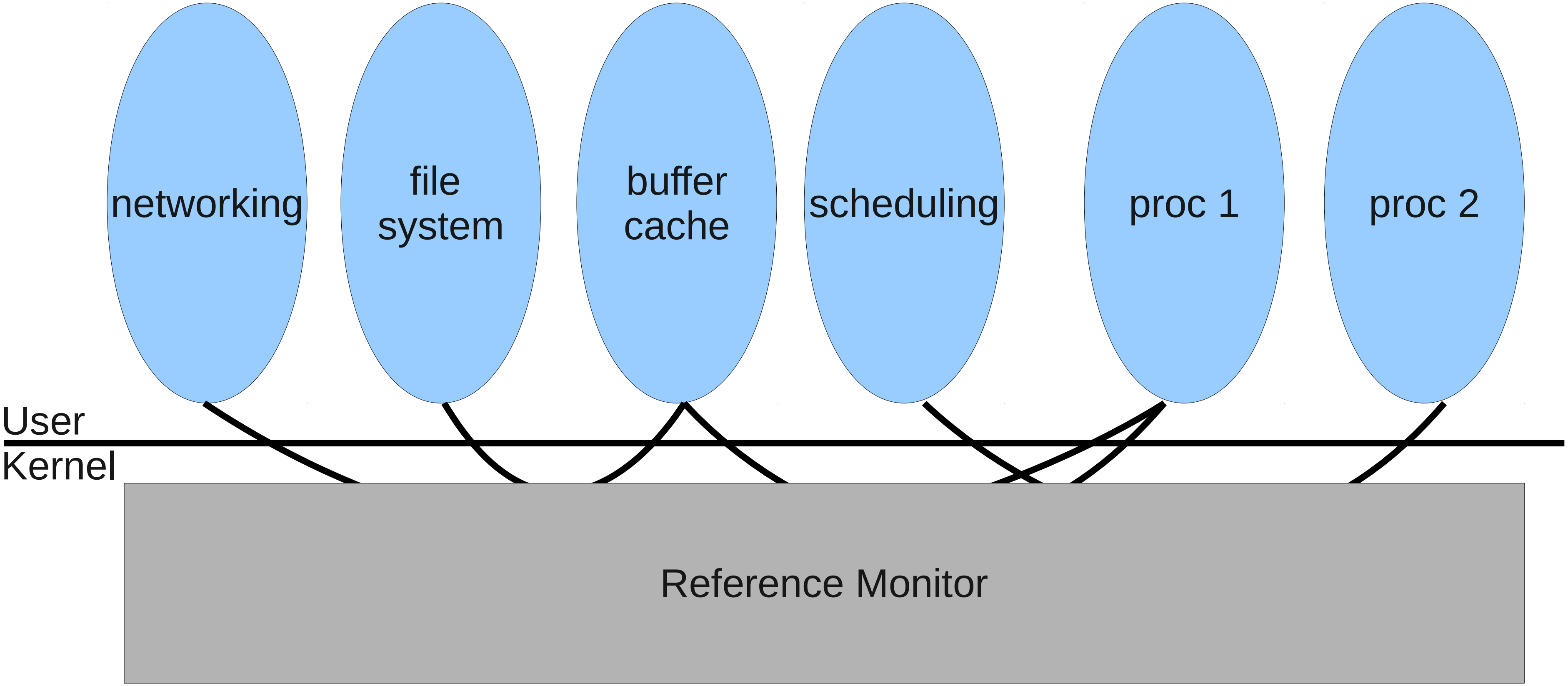


- Roles meant to correspond to natural concepts in an organization

Secure OSes

- Must ensure integrity of the reference monitor
  - Must be implemented somewhere, typically in kernel
  - Must ensure integrity of the whole kernel!
  - Trusted Code Base (TCB) – all code on the system that must be trusted to ensure correctness of protection mechanisms and policies

Secure Oses II



Questions

- Can the user be a good reference monitor?
  - User Account Protection (UAP)
  - App stores installation process
- Can Windows/Linux/OS X every be secure?
- Why don't we separate all system resources so no users can access the same resources?
  - How about this separation for processes?
  - No information flow between users!